

# Boxfill File

```
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilitizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to manipulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#       the system you're on.
# sys - This module provides access to some objects used or maintained by
#       the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

# Initial VCS:
v = vcs.init()

# Show the list of persistent boxfill graphics methods.
v.show('boxfill')

print "Press the Return key to see next plot."
sys.stdin.readline()

# Assign the variable "bf_asd" to the persistent 'ASD' boxfill graphics methods.
bf_asd = v.getboxfill( 'ASD' )

# Plot the data using the above boxfill graphics method.
v.plot( data, bf_asd )

print "Press the Return key to see the 'ASD' boxfill graphics methods attributes."
sys.stdin.readline()

# List the 'ASD' boxfill graphics methods attributes.
bf_asd.list()

print "Press the Return key to see next plot."
sys.stdin.readline()

v.clear()
# Change 'ASD' boxfill graphics methods attributes by entering the following commands.
bf_asd.level_1 = 20          # set the minimum data value
bf_asd.level_2 = 80          # set the maximum data value
bf_asd.legend = {25:'Blue', 40:'Green', 58:'Yellow', 77:'Red'}
bf_asd.color_1 = 40           # change the 1st color index
bf_asd.color_2 = 228          # change the 2nd color index
bf_asd.datawc(-45.0, 45.0, -90.0, 90.0)   # change the region
v.plot( data, bf_asd )

print "Press the Return key to see next plot."
sys.stdin.readline()

v.clear()
# Generate a log10 boxfill graphics method plot
bf_asd.boxfill_type='log10'  # change the boxfill type to log10
bf_asd.datawc(1e20,1e20,1e20,1e20) # change to default region
bf_asd.level_1=1e20           # change to default minimum level
```

```

bf_asd.level_2=1e20          # change to default maximum level
bf_asd.color_1=16             # change 1st color index value
bf_asd.color_2=239            # change 2nd color index value
v.plot( data, bf_asd )

print "Press the Return key to see next plot."
sys.stdin.readline()

v.clear()
# Generate a custom boxfill graphics method plot
bf_asd.boxfill_type='custom'  # change the boxfill type to custom
bf_asd.levels=(0,20,35,40,75,100) # set the custom ranges
bf_asd.fillareacolors=(16,55,155,200,235) # set the color indices
v.plot( data, bf_asd )

print "Press the Return key to see next plot."
sys.stdin.readline()

# Create a persistent boxfill graphics methods from an existing boxfill graphics method.
bf_new = v.createboxfill( 'new', 'ASD' ) # create new from ASD
bf_new2 = v.createboxfill( 'new2','quick')# create new2 from quick
bf_new.color_1=50               # change color level
bf_new.list()                  # list its attributes
v.show('boxfill')              # show boxfill list with new and new2
v.removeobject( bf_new )        # remove new from boxfill list
v.show('boxfill')              # show boxfill list without new

print "Press the Return key to quit."
sys.stdin.readline()

```